J. Symbolic Computation (1991) **12**, 459–474

Fast Recognition of Doubly Transitive Groups

P. J. CAMERON AND J. CANNON†

School of Mathematical Sciences, Queen Mary and Westfield College, Mile End Road, London E1 4NS, UK
† *School of Mathematics and Statistics, The University of Sydney N.S.W. 2006, Australia*

(*Received 9 September* 1990)

The availability of the classification of finite simple groups allows us to design algorithms for identifying the composition factors of finite groups. This paper presents an algorithm which identifies any finite doubly transitive permutation group G. If we exclude the 2-transitive subgroups of the one-dimensional affine group and 14 small exceptional groups, the cost of our algorithm is essentially the cost of constructing a base and strong generating set for G. Consequently, our algorithm avoids the need to compute the soluble residual of G as required by Kantor's composition factors algorithm for a general permutation group.

1. Introduction

The recently completed classification of finite simple groups (Gorenstein, 1982) opens up the possibility of designing computer programs which, given some concrete representation of a finite group, will produce a description of the abstract structure of the group. Ideally, we would like to be able to name the composition factors of the group and specify how they act on one another. Unfortunately, the theory of group extensions is not yet sufficiently advanced so as to make this an attainable goal for an arbitrary group.

In this paper we describe a simple and fast algorithm which, given a set of generating permutations for a doubly transitive permutation group G, will produce a description of the structure of G. Neumann (1987) and Kantor (1991) describe algorithms which are capable of naming the composition factors of an arbitrary permutation group. In the case of doubly transitive groups, both the Neumann and Kantor algorithms require the computation of the derived series. Our algorithm avoids the computation of the derived series other than in the case of a few very small groups (2-transitive subgroups of $A\Gamma L(1, q)$, $PSL(2, 7)$ and 13 exceptional groups discovered by Huppert (1957) and Hering (1974)). Except in the case of the above groups, the only significant group theoretic computation required is the construction of a base and strong generating set. Hence we believe that the running time of our algorithm approaches the best possible for a deterministic recognition algorithm.

The class of doubly transitive groups is particularly important since it includes the symmetric and alternating groups in their natural representations, and "geometric" groups such as the projective group $PGL(n, q)$ and the affine group $AGL(n, q)$. Since a general recognition procedure for permutation groups will detect double transitivity at an early stage (once a base and strong generating set is known), it is straightforward to incorporate our algorithm into a general procedure.

This research was supported by the Australian Research Council.

0747-7171/91/100459+16 $03.00/0

Before turning to the detail of the algorithm, it is necessary to discuss the type of description such an algorithm could reasonably be expected to produce. As is well-known, the classification of finite simple groups, together with results of Maillet (1895), Curtis, Kantor & Seitz (1976), Howlett (1974), Huppert (1957), Hering (1974), and many other mathematicians, has led to a classification of all finite doubly transitive groups (see Cameron, 1981). We have tabulated these groups, together with additional information that we require about them, in the next section. Note, however, that the table is not a list of individual groups. For example, suppose we have discovered that our group G has a normal subgroup $N = PSL(d, q)$, where $q = p^e$, with p a prime. Then G is contained in the normalizer of N in the symmetric group, namely $P\Gamma L(d, q)$, and the complete description of G requires specification of G/N, a subgroup of $X = P\Gamma L(d, q)/PSL(d, q)$. Now X is a metacyclic group generated by a group of order $(d, q-1)$ consisting of diagonal automorphisms, and a group of order e consisting of field automorphisms. The specification of a subgroup of X may be trivial (if $X = 1$) or quite complicated. Rather than having the algorithm provide a description of an arbitrary subgroup of X, we content ourselves with having it report certain extreme cases: for example, whether G is $PSL(d, q)$ or $P\Gamma L(d, q)$. This can easily be done by comparing the order of G with those of $PSL(d, q)$ and $P\Gamma L(d, q)$.

Our algorithm is able to recognize most types of doubly transitive group solely on the basis of number-theoretic conditions on the degree, order, and orbit lengths of a two-point stabilizer (orbit lengths of a three-point stabilizer when G is triply transitive). In cases where the number-theoretic criteria fail to distinguish between different groups (such as $PSL(2, 7)$ and $A\Gamma L(1, 8)$, or any two sharply 2-transitive groups) we resort to using information about the internal structure of the group (the derived group, or the derived length). However, the groups for which this additional information is needed will always be relatively small so that the cost of obtaining this information will not be excessive.

Throughout this paper we shall assume that G is a doubly transitive permutation group acting faithfully on the set Ω. We further assume that G is given in terms of a set of generating permutations. Given a base and strong generating set for G (Cannon, 1984*b*), the order of G, and the orbits of the pointwise stabilizer of a sequence of points from Ω may be obtained at little cost. Such a base and strong generating set may be constructed through application of some variant of the Sims-Schreier algorithm (Sims, 1971; Leon, 1980; Brownie, Cannon & Sims, in preparation). Once a base and strong generating set are known for G, it is a simple matter to read off the degree of transitivity of G, and consequently, to determine whether or not G is the alternating group A_n, or the symmetric group S_n. However, if it does happen that G is A_n or S_n, the Sims-Schreier algorithm will be very expensive. Consequently, we begin by applying a fast probabilistic test for A_n and S_n before resorting to the Sims-Schreier algorithm (see Cameron & Cannon, in preparation). In the remainder of this paper we shall be concerned with the recognition of a doubly transitive group that is neither A_n nor S_n. For most doubly transitive groups, the running time of our algorithm will be entirely dominated by the time needed to construct a base and strong generating set.

Finally, we observe that our algorithm presupposes the correctness of the classification of finite simple groups. However, the legacy of pre-classification permutation group theory is a large supply of characterization theorems for particular classes of doubly transitive groups. If there are philosophical or practical objections to using the classification, then our algorithm can be employed to produce a "tentative" identification. This identification can then be confirmed by verifying that the hypothesis of an appropriate characterization

theorem holds for the given group. If either the algorithm fails to recognize the group, or the subsequent check fails, then a "new" 2-transitive group has been found (or, more likely, the algorithm or its implementation has been shown to be incorrect). In fact the algorithm is well-adapted to checking geometric characterizations: for example, we find a line in the geometry of a projective or affine group. We hope to return to this matter subsequently.

2. Orbit Lengths and Breakpoints

Table 1 gives a list of the doubly transitive groups, other than the symmetric and alternating groups, together with information about the orbit lengths of their two-point stabilizers. This information is not complete. For example, any subgroup of $P\Gamma U(3, q)$ containing $PSU(3, q)$ acts as a 2-transitive group on the q^3+1 points of the associated unital. However, we do not need to know the orbit lengths for all such subgroups as, at the point in the algorithm where these groups are recognized, the partial information in the table will be sufficient. For the families with numbers 1–16, the groups are specified by a particular normal subgroup N (not always the minimal normal subgroup); then G is contained in the normalizer of N in the symmetric group. Families 17 and 18 are taken from lists compiled by Huppert (1957) and Hering (1974), respectively.

We have slightly rearranged Hering's list. For the three groups $T.A_7$, $T.A_6$ and $T.PSU(3, 3)$, the orbit lengths of the 2-point stabilizer are the same as for their overgroups $T.GL(4, 2)$, $T.Sp(4, 2)$ and $T.G_2(2)$ respectively, so we have moved them to the appropriate point in the table.

Let G be a 2-transitive group of degree n, in which the orbit lengths of a 2-point stabilizer $G_{\alpha\beta}$, in non-decreasing order, are $n_1, n_2, \ldots, n_d$ (where $n_1 = n_2 = 1$). A number r such that $2 < r < d$ is said to be a *breakpoint* if

$$\left(\sum_{i=1}^{r} n_i - 1\right)^2 \leq n_{r+1}.$$

The *cumulative set* associated with a breakpoint r is the union of the $G_{\alpha\beta}$-orbits of lengths $n_1, \ldots, n_r$ (this is well-defined, even though the sequence of orbits is not).

We remark that there is no *a priori* theoretical reason for the precise definition of a breakpoint that we have chosen. The motivation is that the "geometric" 2-transitive groups (notably projective and affine groups) are distinguished by the fact that the stabilizer of two points fixes the line through those points (which consists of relatively small orbits), while points off this line lie in large orbits. The particular definition was guided by a desire to have, as far as possible, a clean separation between classes of comparable complexity; the extent to which we succeeded is indicated by the next result.

LEMMA 2.1. (i) *All groups of types* 1, 9, 10, 11 *have breakpoints; in each case, the cumulative set associated with the last breakpoint is a line of the projective or affine space.*

(ii) *Groups of types* 15, 16, 17, 18 *may or may not have breakpoints.*

(iii) *No group of any other type has a breakpoint.*

PROOF. This result is largely verified by inspection of the table. The assertion for types 1, 9, 10, 11 is clear.

Consider, as an example, type 16, in the case where $q = p^m$, with p and m prime. If $G = A\Gamma L(1, q)$, then $G_{\alpha\beta}$ has p fixed points and $(q-p)/m$ orbits of length m; so it has

Table 1. Doubly transitive groups

Type	Description (normal subgroup for types 1–15)	Degree n	$G_{\alpha\beta}$-orbit lengths
1	$PSL(d, q)$, $q \geq 3$ $A_7(d=4, q=2)$	$\frac{q^d-1}{q-1}$	$1, 1, q-1, q^2\left(\frac{q^{d-2}-1}{q-1}\right)$
2	$PSL(2, q)$	$q+1$	$1, 1, q-1$, or $1, 1, \frac{1}{2}(q-1), \frac{1}{2}(q-1)$
3	$Sz(q)$, $q=2^{2d+1}$, $d \geq 1$	q^2+1	$1, 1, q-1$, multiples of $q-1$
4	$PSU(3, q)$, $q>2$	q^3+1	$1, 1, q-1$, multiples of q^2-1 or $\frac{1}{3}(q^2-1)$
5	$R(q)$, $q=3^{2d+1}$, $d>1$ $P\Gamma L(2, 8)$ $(q=3)$	q^3+1	$1, 1, \frac{1}{2}(q-1), \frac{1}{2}(q-1)$, multiples of $q-1$
6	$Sp(2d, 2)$, $d \geq 3$	$2^{d-1}(2^d \pm 1)$	$1, 1, 2(2^{d-2} \pm 1)(2^{d-1} \mp 1)$, $2^{2(d-1)}$
7	Sporadic groups: 3-transitive		$1, 1, n-2$
	M_{11}	11	
	M_{12}	12	
	M_{11}	12	
	M_{22}, Aut(M_{22})	22	
	M_{23}	23	
	M_{24}	24	
8	Sporadic groups: not 3-transitive		
	$PSL(2, 11)$	11	1, 1, 3, 6
	HS	176	1, 1, 12, 72, 90
	Co_3	276	1, 1, 112, 162
9	$T.SL(d, q)$, $d>2$, $q=p^e>2$,	q^d	1, 1, divisors of e summing to $q-2$, q^d-q
10	$T.Sp(d, q)$, $q=p^e>2$, d even	q^d	1, 1, divisors of e summing to $q-2$, $q^{d-1}-q$, multiples of q^{d-1} summing to q^d-q^{d-1}
11	$T.G_2(q)$, $q=2^e>2$	q^6	1, 1, multiples of e summing to $q-2$, q^3-q, q^5-q^3, multiples of q^5 summing to q^6-q^5
12	$T.SL(d, 2)$ $T.A_7$ $(d=4)$	2^d	$1, 1, 2^d-2$
13	$T.Sp(d, 2)$, d even $T.A_6$ $(d=4)$	2^d	$1, 1, 2^{d-1}-2, 2^d-2^{d-1}$
14	$T.G_2(2)$ $T.PSU(3, 3)$	2^6	1, 1, 6, 24, 32
15	$T.SL(2, q)$, $q=p^e>3$	q^2	1, 1, divisors of e summing to $q-2$, multiples of q summing to q^2-q
16	$G \leq A\Gamma L(1, q)$, $q=p^e>4$	q	1, 1, divisors of e
17	Soluble exceptions (Huppert)	3^2	1, 1, 1, 6 or 1, 1, 1, 3, 3
		5^2	divisors of 4
		7^2	divisors of 2
		11^2	divisors of 2
		23^2	all 1
		3^4	divisors of 8
18	Insoluble exceptions (Hering)	3^4	divisors of 48
		3^6	divisors of 3
		3^4	divisors of 2
		11^2	divisors of 2
		19^2	divisors of 2
		29^2	divisors of 2
		59^2	divisors of 2

a breakpoint if and only if $m \geq (p-1)^2$. Similar remarks apply to subgroups of $A\Gamma L(1, q)$. For other groups with a regular normal subgroup (9, 10, 11 and 15), the group induced on an affine line is a subgroup of $A\Gamma L(1, q)$, and so may itself have breakpoints; this is why part (i) of the lemma refers to the last breakpoint. For type 15, also, there may or may not be an additional breakpoint whose cumulative set is a line of the affine plane; this is the case, for example, if $G = T.GL(2, q)$, but not for any proper subgroup.

To complete the proof, the only non-trivial cases are those involving $PSU(3, q)$, $Sz(q)$ or $R(q)$, possibly with diagonal (in the first case) or field automorphisms adjoined. It suffices to observe that the group of field automorphisms has order at most $2 \log q$ (for $PSU(3, q)$) or $\log q$ (for the others), and that it fixes at least one long orbit (an orbit containing points defined over the prime subfield).

This discussion suggests that the overall structure of our algorithm should be as follows:

PROCEDURE doubly transitive (G)

Input: *a 2-transitive group G of degree n*

Output: *a description of the group G*

if G is the alternating or symmetric group of degree n then
  print a message to this effect and exit;
end if;
if G has a breakpoint then
  breakpoints (G; *recognized*);
else
  no_breakpoints (G; *recognized*);
end if;
if recognized is false then
  exceptions (G);
end if;

end.

Procedure *breakpoints* recognizes groups of types 1, 9–11, procedure *no_breakpoints* recognizes types 2–8, 12–14, and procedure *exceptions* recognizes types 15–18. The Boolean variable *recognized* appearing in the parameter list for procedures *breakpoints* and *no_breakpoints* is set true by the corresponding procedure if the group G is recognized by that procedure.

One more piece of information about breakpoints is useful in the test for types 1 and 9–11.

LEMMA 2.2. *Let G be a 2-transitive group of degree n having a breakpoint whose cumulative set has size m. Then n is a power of m if and only if G has a regular normal subgroup.*

PROOF. If G does not have a regular normal subgroup, then it is of type 1, with $n = (q^d - 1)/(q-1)$ and $m = q+1$, for some $d \geq 3$. By the theorem of Zsigmondy (1892), unless $q = 2$ and $d = 6$, $q^d - 1$ has a primitive prime divisor r, which does not divide either $q-1$ or $q+1$; so n is not a power of m. Furthermore, 63 is not a power of 3.

Conversely, suppose that G is of one of the types 9-11 or 15-18. By inspection, the result holds for types 17 and 18, so we ignore these. If the cumulative set is an affine line, the result also holds. In any other case, the cumulative set is a subset of an affine line, and it suffices to prove the result in the case $G \geq A\Gamma L(1, q)$.

In this case, G_{01} consists of field automorphisms; let its fixed field be, say, $GF(r)$, so that $q = r^x$ for some x. Let y be the length of the orbit following the breakpoint. For any divisor u of x, the union of all orbits of length dividing u is $GF(r^u)$; so the cumulative set is a union of subfields. Suppose that it is not a single subfield. Then it contains two subfields, say $GF(r^u)$ and $GF(r^v)$, whose intersection is a proper subfield of each. Thus, $y \geq r^{2u}$ and $y \geq r^{2v}$, that is, $u, v \leq \frac{1}{2}\log_2 y$, and $uv \leq \frac{1}{4}(\log_2 y)^2 < y$. So the least common multiple z of u and v is less than y, whence $GF(r^z)$ is contained in the cumulative set and contains both $GF(r^u)$ and $GF(r^v)$. We conclude that the cumulative set is a subfield, and the result holds.

Lemma 2.2 leads to the following procedure for the recognition of 2-transitive groups of types 1, 9, 10 and 11:

PROCEDURE breakpoints (G; *recognized*)

Input: *a 2-transitive group G that possesses a breakpoint*

Output: *a description of the group G, if it has been recognized, and a Boolean variable "recognized" which is set true if G is recognized and false otherwise*

recognized := *false*;
n := *degree of* G;
m := *size of the cumulative set*;

{*G does not possess a regular normal subgroup*}

if n is not a power of m then
  $q := m - 1$;
  $d := \log_q((q-1)n+1)$;
  if $q = 2$ *and* $d = 4$ *and* $|G| = 2^3 \cdot 3^2 \cdot 5 \cdot 7$ *then*
    G is A_7;
  else
    G contains $PSL(d, q)$ *as a normal subgroup*;
  end if;
  recognized := *true*;

{*G contains a regular normal subgroup*}

else

  $q := m$;
  $d := \log_q n$;
  r := *last breakpoint of* G;
  let $n_1 \leq \cdots \leq n_r \leq \cdots \leq n_s$ *denote the lengths of the orbits of* $G_{\alpha\beta}$;

  if $d > 2$ *and* $s = r+1$ *then*
    G contains $T.SL(d, q)$ *as a normal subgroup*; *recognized* := *true*;
  else if $n_{r+1} = q^{d-1} - q$ *and* $q^{d-1} | n_i$ *for* $i = r+2, \ldots, s$, *then*
    G contains $T.Sp(d, q)$ *as a normal subgroup*; *recognized* := *true*;

else if $d = 6$ *and* $n_{r+1} = q^3 - q$, $n_{r+2} = q^5 - q^3$, *and* $q^5 | n_i$ *for* $i = r+3, \ldots, s$ *then*
G contains T. $G_2(q)$ *as a normal subgroup*; *recognized* := *true*;
end if;

end if;

end.

The justification of procedure *breakpoints* involves checking that the conditions on orbit lengths which have been used for testing cases 9–11 cannot hold in any of the cases 15–18. The easiest way to see this is to observe that if any of these tests succeeds, then the last orbit length is at least q^{d-1}, where $n = q^d$ and $d \geq 3$; few of cases 15–18 have an orbit length as large as this. (For type 15 this would require $\log_2 n \geq n^{2/3}$, which is never satisfied; for 17 and 18, the degree is a higher power than the second only for $n = 3^4$ or 3^6, which are readily checked. Type 16 can only arise if $n = r^2$, with $r \log_2 r \geq r^{3/2}$; the only solution is $n = 16$, $G = T.\Sigma L(2, 4)$, with orbit lengths 1, 1, 2, 4, 8 for $G_{\alpha\beta}$, but this does not in fact have a breakpoint.)

3. The No-breakpoint Case

The recognition of non-exceptional groups without breakpoints divides naturally into two parts depending upon whether or not G is 3-transitive (no 3-transitive group possesses a breakpoint). The 3-transitive groups (types 7, 12 and some cases of 2) have to be distinguished using the orbit lengths of a 3-point stabilizer rather than those of a 2-point stabilizer.

PROCEDURE no_breakpoints (*G*; *recognized*)

Input: *a 2-transitive group that does not possess any breakpoints*

Output: *a description of the group G, if it has been recognized, and a Boolean variable "recognized" which is set true if G is recognized by this procedure, and false otherwise*

recognized := *false*;
n := *degree of G*;

{*G is 3-transitive*}

if G is 3-transitive then

let l be the list of orbit lengths of a 3-point stabilizer,
arranged in non-decreasing order;

recognized := *true*;

if $l = (1, 1, 1, 1, n-4)$ *then*
if $n = 16$ *and* $|G| = 2^7 \cdot 3^2 \cdot 5 \cdot 7$ *then*
G is T. A_7;
else
G is T. $GL(d, 2)$;
end if;
else if $n = 11$ *then*
G is M_{11};

else if $n = 12$ *and* $l = (1, 1, 1, 9)$ *then*
  G *is* M_{12};
else if $n = 12$ *and* $l = (1, 1, 1, 3, 6)$ *then*
  G *is* M_{11};
else if $n = 22$ *then*
  if $|G| = 2^7 . 3^2 . 5 . 7 . 11$ *then*
    G *is* M_{22};
  else
    G *is* $Aut(M_{22})$;
  end if;
else if $n = 23$ *then*
  G *is* M_{23};
else if $n = 24$ *and* $l = (1, 1, 1, 21)$ *then*
  G *is* M_{24};
else
  G *contains* $PSL(2, q)$ *as a normal subgroup*;
end if;

{G *is not* 3*-transitive*}

else

let $n_1 \leq n_2 \leq \cdot \cdot \cdot \leq n_s$ *denote the lengths of the orbits of* $G_{\alpha\beta}$;
$l := (n_1, \ldots, n_s)$;

if $l = (1, 1, (n-2)/2, (n-2)/2)$ *and either* $n \neq 8$ *or* G *is perfect then*
  G *contains* $PSL(2, q)$ *as a normal subgroup*; *recognized* := *true*;
else if $n = q^2 + 1$, *for a prime power* q, *and* $n_3 = q - 1$ *then*
  G *contains* $Sz(q)$ *as a normal subgroup*; *recognized* := *true*;
else if $n = q^3 + 1$, *for a prime power* q, *and* $q^3 > 2$ *then*
  if $n_3 = q - 1$ *then*
    G *contains* $PSU(3, q)$ *as a normal subgroup*; *recognized* := *true*;
  else if $n_3 = (q-1)/2$ *then*
    G *contains* $R(q)$ *as a normal subgroup*; *recognized* := *true*;
    {*note that this includes the case when* G *is* $P\Gamma L(2, 8)$ *with* $q = 3$}
  end if;

else if $n = 2^{d-1}(2^d \pm 1)$ *for some integer* d, *and* $n \geq 28$ *then*
  G *is* $Sp(2d, 2)$; *recognized* := *true*;
else if $n = 11$ *and* $n_3 = 3$ *then*
  G *is* $PSL(2, 11)$; *recognized* := *true*;
else if $n = 176$ *then*
  G *is* HS; *recognized* := *true*;
else if $n = 276$ *then*
  G *is* Co_3; *recognized* := *true*;
else if $n = 2^d$, *for some integer* d, *and* $n_3 = 2^{d-1} - 2$ *then*
  if $d = 4$ *and* $|G| = 2^7 . 3^2 . 5$ *then*
    G *is* $T . A_6$;
  else
    G *is* $T . Sp(d, 2)$,

```
      end if;
      recognized := true;
    else if n = 2^6 and l = (1, 1, 6, 24, 32) then
      if |G| = 2^11 . 3^3 . 7 then
        G is T. PSU(3, 3);
      else
        G is T. G_2(2);
      end if;
      recognized := true;

    end if;

  end if;

  end.
```

For justification, the degrees of the remaining groups are
(a) $q+1$ (type 2)
(b) q^2+1, $q=2^{2d+1}$, $d>1$ (type 3)
(c) q^3+1, $q>2$ (types 4, 5)
(d) $2^{d-1}(2^d \pm 1)$ (type 6)
(e) 11, 176, 276 (type 8)
(f) 2^d (types 13, 14)
(g) q (types 15–18)
where q denotes a prime power. What overlaps can occur? Clearly (b) and (c) are special cases of (a), but inspection of n_3 distinguishes the groups. We show in Appendix 1 that the only overlaps between (d) and (a) are 6, 10 (which are irrelevant here, since $Sp(4, 2) = S_6 = P\Sigma L(2, 9)$) and 28 (where our algorithm is correct since $Sp(6, 2)$ has $n_3 = 10$, while $PSL(2, 27)$, $PSU(3, 3)$ and $P\Gamma L(2, 8)$ have $n_3 = 13, 2, 1$ respectively). There is no overlap between (e) and any other type, except that 11 is a prime power; but $A\Gamma L(1, 11)$ has $n_3 = 1$. As before, (f) is correctly distinguished from (g). The only troublesome overlap is between (a)-(c) and (g). In fact, (b) is never a prime power since, when $q = 2^{2d+1}$, we have the factorization $q^2+1 = (q+2^{d+1}+1)(q-2^{d+1}-1)$, and the factors are coprime. Similarly, $q^3+1 = (q+1)(q^2-q+1)$ is a prime power only when $q = 2$. Confusion between 2 and 16 requires $\log_2 n \geq \frac{1}{2}(n-2)$, which only occurs for $n = 8$; here $PSL(2, 7)$ and $A\Gamma L(1, 8)$ are indistinguishable on the basis of order or orbit lengths, so we must invoke a group-theoretic test. Finally, type 2 cannot be confused with types 15, 17 or 18 by our algorithm.

4. Handling the Exceptions

Any group not already identified must be of type 15, 16, 17 or 18. The groups of type 17 may be recognized with the aid of the following lemma:

LEMMA 4.1. *Let G be a 2-transitive group of degree n. Suppose*
(a) *n is a square greater than 9, say $n = q^2$; and*
(b) *some initial segment of the list of $G_{\alpha\beta}$-orbit lengths has sum q, and that all subsequent orbit lengths are multiples of q.*

Then, either G contains $T.SL(2, q)$ as a normal subgroup, or the $G_{\alpha\beta}$-orbit lengths are 1, 1, 2, 4, 4, 4, *and G is $A\Gamma L(1, 16)$.*

PROOF. Clearly, if $T.SL(2, q) \lhd G$, then G satisfies conditions (a) and (b). By inspection, no group of type 17 or 18 satisfies these conditions. Suppose that $G \leq A\Gamma L(1, \bar{q})$, where G satisfies conditions (a) and (b). Let $\bar{q} = q^2$. Then $\log_2(q^2) \geq q$, whence $q \leq 4$, and so $\bar{q} \leq 16$. The only possibility for G is $A\Gamma L(1, 16)$, with $G_{\alpha\beta}$-orbit lengths 1, 1, 2, 4, 4, 4. However, no group G with $T.SL(2, q) \lhd G$ has these orbit lengths. For the orbit lengths in $T.SL(2, 4)$ itself are 1, 1, 1, 1, 4, 4, 4; if we adjoin a scalar or field automorphism, then some amalgamation of orbits of length 4 must occur. So, if G satisfies the hypotheses of the lemma and does not have $G_{\alpha\beta}$-orbit lengths 1, 1, 2, 4, 4, 4, then $T.SL(2, q) \lhd G$.

PROCEDURE exceptions (G)

Input: a 2-transitive group G that may or may not possess breakpoints and which does not satisfy any of the criteria of procedures breakpoints and no_breakpoints

Output: a description of the group G

$n :=$ degree of G;
let $n_1 \leq n_2 \leq \cdots \leq n_s$ denote the lengths of the orbits of $G_{\alpha\beta}$;
$l := (n_1, n_2, \ldots, n_s)$;

if n is a square greater than 9, say $n = q^2$, and there exits an integer m, $1 \leq m < s$, such that $\sum_{i=1}^{m} n_i = q$, and $q | n_j$, $j = m+1, \ldots, s$, and $l \neq (1, 1, 2, 4, 4, 4)$ then G has $T.SL(2, q)$ as a normal subgroup;
end if;

if G is not soluble then
G is one of the Hering exceptions (type 18);
else
$d :=$ derived length of G;
$q := n$;
if $d = 2$ then
G is $AGL(1, q)$;
else if $d = 3$ then
G is a subgroup of $A\Gamma L(1, q)$, but is not $AGL(1, q)$;
else if $d = 4$ or $d = 5$ then
G is one of the Huppert exceptions (type 17);
end if;
end if;

end.

Lemma 4.1 justifies the recognition of groups of type 15. Since the only insoluble groups that can reach this point are the Hering exceptions, groups of type 18 will be correctly recognized. In all of Huppert's examples, $G_0 \cong G/T$ involves $SL(2, 3)$, with derived length 3, so G has derived length at least 4. On the other hand, $A\Gamma L(1, q)$ has derived length 3, with derived series

$$A\Gamma L(1, q) \rhd AGL(1, q) \rhd T \rhd 1,$$

if q is not prime; if q is prime, then $A\Gamma L(1, q) = AGL(1, q)$ and the first term is absent. If $G \leq A\Gamma L(1, q)$, and G has derived length 2, then G_0 is an irreducible abelian group of automorphisms of T, so its endomorphism ring is $GF(q)$, and $G \leq AGL(1, q)$.

We may, if desired, complete the "exceptions" procedure by assigning to a group of type 17 or 18, its number in the paper of Huppert (1957) or Hering (1974). (Note that Huppert lists individual groups, while Hering's classification is somewhat coarser.)

5. Implementation and Performance

The algorithm described above has been implemented in the algebraic programming language Cayley (Cannon, 1984*a*). Since the Cayley language contains standard functions for computing a base and strong generating set, stabilizers, orbits and derived series, the coding and debugging of our program took under a day. The recognition program is distributed to users of the Cayley system as file DTGROUPS in the Cayley library CAYPROC.

We now consider the running time of the algorithm. There are four computationally significant steps:

(i) Testing whether G is the alternating or symmetric group;
(ii) Constructing a base and strong generating set for the group G;
(iii) Computing the orbits of a two- or three-point stabilizer in G;
(iv) Computing the derived series in the cases where G is a 2-transitive subgroup of $A\Gamma L(1, q)$, $PSL(2, 7)$, one of the 6 Huppert soluble exceptions, or one of the 7 Hering non-soluble exceptions.

We examine each of these steps in detail. A Las Vegas style probabilistic algorithm is employed to recognize A_n or S_n (Cameron & Cannon, in preparation). If G is the alternating or symmetric group, there is a very high probability that the algorithm will quickly produce a proof of this fact. Empirically, we have observed that, on average, this algorithm has running time $O(n)$.

If the alternating/symmetric test has not established that G is alternating or symmetric after a predetermined amount of effort, some form of the Sims–Schreier algorithm is invoked to construct a base and strong generating set for G. This settles the issue definitively and represents the next step in the computation when G is neither A_n nor S_n. Because of the enormous cost involved in the construction of a base and strong generating set for A_n or S_n in the case of large n, the probabilistic test has to be tuned so as to have a very small probability of failing to recognize these groups. Thus, the probabilistic A_n/S_n test introduces some overhead when G is neither of these groups. However, this overhead is usually small compared to the cost of constructing a base and strong generating set.

The original Sims–Schreier algorithm (Sims, 1970) has running time $O(n^6)$, while a version using alternative data structures has running time $O(n^5)$ (Jerrum, 1986). While the average running time for the Sims–Schreier algorithm is much better than these estimates, nevertheless, its application is rarely practical in the case of groups having degree greater than 1000. For groups of larger degree a different approach must be adopted. The basic technique involves the construction of a base and strong generating set (or an approximation) using a probabilistic algorithm. A separate algorithm is then used to verify the correctness or otherwise of the putative base and strong generating set. An early verification algorithm based on the Todd–Coxeter procedure is described by Leon (1980). A new verification algorithm applicable in some cases to groups having degree up to a million is described in Brownie, Cannon & Sims (in preparation).

Table 2. Performance of the 2-transitive group recognition algorithm

Group	Degree	Order									A_n/S_n time T_1	BSGS time T_2	Analysis time T_3	Total time $T = T_1 + T_2 + T_3$
$PGL(4, 8)$	585					2^9	3^3	5^6	13	31	1·0	40	0·5	41·5
$PSL(3, 32)$	1057				2^{15}	3	7	11	31^2	151	2·6	91	0·1	93·7
$PGL(6, 5)$	3906		2^{14}	3^4	5^{15}	7	11	13	31^2	71	7·9	1193	3·4	1204
$PGL(7, 4)$	5461	2^{42}	3^8	5^3	7^2	11	13	17	31	43	11·7	1935	7·6	1954
$PGL(5, 9)$	7381		2^{15}	3^{20}	5^2	7	11^2	13	41	61	17·6	1110	6·4	1134
$PSL(7, 5)$	19531	2^{16}	3^4	5^{21}	7	11	13	31^2	71	19531	7·0	10946	23·3	10976
$PSU(3, 8^2)$	531						2^9	3^4	7	19	1·0	10·5	1·2	12·7
$PSU(3, 11^2)$	1332					2^5	3^2	5	11^3	37	2·8	67	2·1	72
$PSU(3, 13^2)$	2198					2^4	3	7^2	13^3	157	4·9	39	0·8	45
$PSU(3, 17^2)$	4914					2^6	3^4	7	13	17^3	10·9	130	4·4	145
$PSU(3, 19^2)$	6860					2^5	3^2	5^2	7^3	19	15·4	203	1·9	220
$PSU(3, 23^2)$	12168					2^7	3^2	11	13^2	23^2	37·8	1647	8·2	1693
$T.SL(4, 4)$	256					2^{20}	3^4	5^2	7	17	0·6	25·0	0·3	25·9
$T.SL(5, 4)$	1024			2^{30}	3^5	5^2	7	11	17	31	2·4	214	0·9	217
$T.SL(4, 8)$	4096				2^{30}	3^4	5	7^3	13	73	10·0	408	3·0	421
$T.GL(4, 9)$	6561				2^{15}	3^{20}	5^2	7	13	41	14·7	1573	3·6	1591
$T.GL(3, 25)$	15625				2^{10}	3^4	5^{12}	7	13	31	34·0	1651	6·3	1691
$T.GL(5, 7)$	16807				2^{12}	3^6	5^2	7^{15}	19	2801	55·0	5600	10·5	5665
$Sz(8)$	65						2^6	5	7	13	0·2	2·7	0·3	3·2
$Sz(32)$	1025						2^{10}	5^2	31	41	2·4	11·9	1·7	16·0
HS	176					2^9	3^2	5^3	7	11	0·5	9·3	0·2	10·0
Co_3	276				2^{10}	3^7	5^3	7	11	23	0·7	13·7	0·2	14·6

The construction of the orbits of a two- or three-point stabilizer is extremely fast. A set of generators for the stabilizer of any initial segment of the chosen base for G can be immediately identified from among the corresponding strong generators for G. If H is an r-generator permutation group acting on the set Ω of cardinality n, the orbits of H on Ω can be computed in time $O(rn)$.

Although the computation of the derived series for an arbitrary group is rather expensive (being comparable in cost to constructing a base and strong generating set for the group), our algorithm has been carefully designed to avoid the use of the derived series except in the case of a short list of relatively small groups. These groups are $PSL(2, 7)$, $A\Gamma L(1, q)$, the 2-transitive subgroups of $A\Gamma L(1, q)$, the Huppert exceptions, and the Hering exceptions.

Having identified the isomorphism type T of the "critical" normal subgroup N of G (N is typically the soluble residual of G), the appropriate group order formula is applied and the result is compared with the order of G. This enables the program to recognize various extreme situations such as the case $G = T$, or the case $G = Aut(T)$.

The recognition algorithm has been coded as a Cayley procedure (Cannon, 1984*a*). This procedure uses Cayley standard functions (coded in C) for performing the A_n/S_n test, for constructing a base and strong generating set, and for computing orbits. Illustrative run times for a range of 2-transitive groups are presented in Table 2. These timings were obtained using Cayley V3.8 running on a SUN 4/65. The generating sets used were obtained either using an appropriate Cayley function ($PSL(n, q)$, $PGL(n, q)$, $PSU(3, q^2)$, $T.SL(n, q)$, $T.GL(n, q)$) or from the Cayley library PERGPS (HS and Co_3).

Upon inspection of Table 2, it is seen that the execution time for our algorithm is totally dominated by the cost of constructing a base and strong generating set. For

Table 3. Ratios of the execution times for the Kantor algorithm with the 2-transitive algorithm

Group	Degree	Kantor algorithm/ 2-transitive algorithm
$PGL(4, 8)$	585	2·0
$PSL(3, 32)$	1057	1·9
$PGL(6, 5)$	3906	3·5
$PGL(7, 4)$	5461	2·4
$PGL(5, 9)$	7381	2·2
$PSL(7, 5)$	19531	2·0
$PSU(3, 8^2)$	531	1·6
$PSU(3, 11^2)$	1332	1·3
$PSU(3, 13^2)$	2198	2·0
$PSU(3, 17^2)$	4914	1·8
$PSU(3, 19^2)$	6860	1·7
$PSU(3, 23^2)$	12168	1·2
$T.SL(4, 4)$	256	1·9
$T.SL(5, 4)$	1024	1·9
$T.SL(4, 8)$	4096	2·2
$T.GL(4, 9)$	6561	1·8
$T.GL(3, 25)$	15625	2·2
$T.GL(5, 7)$	16807	2·4
$Sz(8)$	65	1·1
$Sz(32)$	1025	1·9
HS	176	1·4
Co_3	276	1·7

purposes of comparison, Table 3 displays, for each group listed in Table 2, the ratio of the execution time of the Kantor algorithm (Kantor, 1991) to the execution time of our 2-transitive group algorithm. For example, the coefficient 2·0 opposite the group $PGL(4, 8)$ in Table 3 indicates that the Kantor algorithm took twice as long as the 2-transitive group algorithm to deduce the names of its composition factors. We observe that the superiority of the 2-transitive algorithm is particularly evident in the case of large degree groups which are not perfect. This reflects the fact that in such groups, the Kantor algorithm has to perform a number of iterations to construct the solvable residual.

6. Conclusion

By carefully analysing the possible orbit structures for a two-point stabilizer in a 2-transitive group (three-point stabilizer in a 3-transitive group), we are able to recognize most 2-transitive groups using only a knowledge of these orbit lengths (occasionally combined with the group order).

The resulting algorithm has the following advantages.

(i) Its cost is essentially the cost of constructing a base and strong generating set (except in the case of a small number of relatively small groups). Hence its running time must be close to being optimal for a deterministic algorithm.

(ii) The algorithm has a particularly simple structure.

Certain classes of group theoretic algorithms perform poorly when applied to a 2-transitive group. The availability of a cheap recognition algorithm for such groups opens up the possibility of employing algorithms which are specifically engineered for each particular family of 2-transitive groups.

References

Butler, G., Cannon, J. J. (1982). Computing in permutation and matrix groups I: Normal closure, commutator subgroups, series. *Math. Comp.* **39**, 633–670.

Cameron, P. J. (1981). Finite permutation groups and finite simple groups. *Bull. London Math. Soc.* **13**, 1–22.

Cannon, J. J. (1984*a*). An introduction to the group theory language Cayley. In: (Atkinson, M., ed.) *Computational Group Theory*. London: Academic Press, pp. 145–183.

Cannon, J. J. (1984*b*). A computational toolkit for finite permutation groups. In: (Aschbacher, M. *et al.*, eds) *Proceedings of the Rutgers Group Theory Year* 1983–1984. Cambridge: Cambridge University Press, pp. 1–18.

Curtis, C. W., Kantor, W. M., Seitz, G. (1976). The 2-transitive representations of the finite Chevalley groups. *Trans. Amer. Math. Soc.* **218**, 1–57.

Gorenstein, D. (1982). *Finite Simple Groups*. New York: Plenum Press.

Hering, C. (1974). Transitive linear groups and linear groups which contain irreducible subgroups of prime power. *Geometriae Dedicata* **2**, 425–460.

Howlett, R. B. (1974). On the degrees of Steinberg characters of Chevalley groups. *Math. Z.* **135**, 125–135.

Huppert, B. (1957). Zweifach transitive, auflösbare permutations-gruppen. *Math. Z.* **68**, 126–150.

Jerrum, M. (1986). A compact representation for permutation groups. *J. Algorithms* **7**, 60–78.

Kantor, W. M. (1991). Finding composition factors of permutation groups of degree $n \leq 10^6$. *J. Symbolic Computation* **12**, 517–526.

Leon, J. S. (1980). On an algorithm for finding a base and strong generating set for a group given by defining permutations. *Math. Comp.* **35**, 941–974.

Maillet, E. (1895). Sur les isomorphes holoédriques et transitifs des groupes symétriques ou alternés. *J. Math. Pures Appl.* **5** (1), 5–34.

Neumann, P. M. (1987). Some algorithms for computing with finite permutation groups. In: (Robertson, E. F., Campbell, C. M., eds) *Groups-St. Andrews* 1985. Cambridge: Cambridge University Press, pp. 59–92.

Sims, C. C. (1971). Computation with permutation groups. In: (Petrick, S., ed.) *Proceedings of the Second Symposium on Symbolic and Algebraic Manipulation, Los Angeles*. Assoc. Comp. Mach., New York.

Zsigmondy, K. (1892). Zur Theorie der Potenzreste. *Monats. fur Math. und Phys.* **3**, 265–284.

Appendix
On the Degrees of 2-Transitive Groups

The degree n of a 2-transitive group G which is not S_n or A_n often carries sufficient information about G, for its recognition (in the sense of this paper). However, there are some awkward overlaps between various families of degrees, and resolution in these special cases seems to require techniques similar to the general methods we have used. For this reason, we chose not to base our algorithm on analysis of n. However, we employed a specific result to identify the groups $Sp(2d, 2)$, whose proof is given here.

A number $n > 4$ is the degree of a 2-transitive group other than the symmetric or alternating group if and only if n is of one of the following forms:

(i) a prime power;
(ii) $(q^e-1)/(q-1)$, q a prime power, $e \geq 2$;
(iii) $2^{d-1}(2^d \pm 1)$, where $d \geq 2$;
(iv) 22, 176 or 276.

In this list, we have minimized overlap subject to easy description. For example, there is no overlap between (i)-(iii) and (iv), or between (i) and (iii). The problem of deciding which numbers fall under (i) and (ii) includes the classical problems of Fermat and Mersenne primes, as well as others such as $3^2 = (8^2-1)/(8-1)$, $31 = (5^3-1)/(5-1) = (2^5-1)/(2-1)$, and $11^2 = (3^5-1)/(3-1)$; we cannot expect a complete description. But the overlap between (ii) and (iii) is easily described:

PROPOSITION A.1. *The only numbers falling under cases* (ii) *and* (iii) *above are* 6, 10 *and* 28.

PROOF. Suppose that $n = (q^e-1)/(q-1) = 2^{d-1}(2^d \pm 1)$, with $d, e \geq 2$ and q a prime power. Then n is even, and so q is odd and e is even. First, we deal with the case $e = 2$, which covers all the possibilities and is all that is needed in our algorithm. In this case we have

$$q = 2^{d-1}(2^d \pm 1) - 1 = (2^{d-1} \pm 1)(2^d \mp 1).$$

Since $2^d \mp 1 = 2(2^{d-1} \pm 1) \mp 3$, the highest common factor of the two factors on the right is at most 3. If it is 1, then $2^{d-1} \pm 1 = 1$, whence $n = 6$; if 3, then $2^{d-1} \pm 1 = 3$, whence $n = 10$ or 28.

So we may suppose that $e > 2$. We use the following fact. Just one of $q-1$ and $q+1$ is divisible by 4; suppose that $2^k \| q \pm 1 (k \geq 2)$ and $2^l \| e$. Then $2^{k+l} \| q^e - 1$.

CASE 1. $q \equiv 1 \pmod 4$. Then $2^k \| q-1$, so

$$2^l \| (q^e-1)/(q-1) = 2^{d-1}(2^d-1),$$

whence $l = d-1$. Then

$$2^{2d} > n > q^{e-1} \geq 5^{2^{d-1}-1},$$

i.e.

$$2d > (2^{d-1}-1)\log_2 5$$

which has no solution for $d > 2$.

CASE 2. $q \equiv -1 \pmod 4$. This time $2 \| q-1$, so

$$2^{k+l-1} \| (q^e-1)/(q-1) = 2^{d-1}(2^d \pm 1),$$

whence $d = k+l$. If $l=1$ then $e \geq 6$ and

$$2^{2(k+1)} > n > q^5 \geq (2^k-1)^5 > 2^{5(k-1)},$$

so $3k<7$, $k=2$, which does not lead to a solution. If $l>1$ then $e \geq 2^l$, and we have

$$2^{2(2k+l)} > n > q^{e-1} \geq (2^k-1)^{2l-1} > 2^{(k-1)(2^l-1)},$$

$$2(k+l) > (k-1)(2^l-1),$$

for which the only possibilities are $l=2$, $k \leq 6$, or $l=3$, $k=2$. In each case q is bounded, and a finite amount of checking now completes the result.